

## 学习报告

### 1. vi 学习报告

由于我一直以来都是用 emacs，之前有接触 vim，但由于最先接触的是 emacs，所以用起 vim 来相当不习惯，vi 可以说是 vim 的简化版，其许多快捷键操作对于写代码非常方便。emacs 和 vi 都是属于常用的编辑器，所以我还是用回了自己习惯的 emacs。

### 2. Java 学习报告

Java 的学习，我在暑假时就已经大致学过 Java 的基本基础的内容。由于学过 C/C++，所以学起 Java 比较容易，Java 的许多语法跟 C++ 非常相似，有些甚至一模一样。跟 C++ 相比，比较不同的一点就是，Java 都是在类里面定义的，即使是 main 函数也是定义在类里面，变量的声明只是声明而已，要想要使用的要用 new 定义。Java 前面的基础内容跟 C++ 的差别不大，但是后面 Java 的 AWT 和 图形界面编程以及 Swing 组件等方面相对与只学过 C 和 C++ 的内容来讲，都是比较新的内容，虽然在 Java 书的时候，在理解方面问题不是很大，自己也有照着书里的例子敲过代码，但是昨天在写 Calculator 的时候，许多语法都还是要通过查找才知道怎么用，所以说对于 Java 的许多内容都还是不熟练，希望接下来能通过这几天的学习，更加熟练使用 Java。

### 3. Ant 学习报告

Ant 的安装与配置比较简单，由于在 linux 下用过一段时间，所以对于命令行也不陌生，所以安装和配置都比较快，配置时设置系统环境变量，修改后保存好后，其实只要用 source 命令，就可以不用重启系统就能测试 Ant 了。Ant 的学习来讲，虽然去年的实训用过 makefile，但是刚开始使用还是老是出错，直接去看 wiki 上的教程比较难，上面的内容比较多，一下子很难消化理解，提供的 pdf 版本的教程相对来讲比较容易学习，对于有学过一点 html 和 makefile 来讲学 Ant 的 xml 格式的比较简单，刚开始用 Ant 测试 helloWorld 时 Build 成功，但由于在 jar 时在修改 manifest 属性时，没有修改 Main-Class 的值，导致编译通过而无法运行，自己尝试修改后才可以运行。发现身边许多同学都出现了这种情况。看来这个问题还是比较普遍的。Ant 的语法来讲相对与 makefile 还是比较好理解，最重要的部分就是 target 部分。现在也大概能看懂 Ant 的 build.xml 文件了。Ant 的测试 helloWorld 的结果如下：

```
Buildfile: build.xml

init:

compile:

dist:
[jar] Building jar: /home/george/GridWorld/project1/h
BUILD SUCCESSFUL
Total time: 0 seconds
```

### 4. Junit 学习报告

Junit 测试工具它的功能有点类似与 gtest 的功能，使用 Junit 就可以实现单元测试，这对于案例后面来讲用处比较打，特别是对于比较大的项目时。Junit 的安装和配置跟 Ant 的安装大同小异，所以也算比较简单。编写测试类时，单独用 Ant 和 Junit 测试 helloWorld 时都能通过但是把两者结合起来时，刚开始出现了许多错误。在使用 Junit 的时候，要先写好测试类，再在编译的时候，加入 Junit 的 jar 包就可以实现测试了，使用 Junit 的命令行如下：

```
javac -cp ../junit-4.9.jar HelloWorldTest.java
java -cp ../junit-4.9.jar -ea org.junit.runner.JUnitCore HelloWorldTest
```

我在测试 HelloWorld 的时候测试结果如下：

```
JUnit version 4.9
.
Time: 0.004
OK (1 test)
```

## 5. Sonar 学习报告

Sonar 是个用于代码质量管理的平台，可以从不同方面检测代码的质量。可以帮助分析我们的代码的复杂度，使代码更加容易让他人理解，还可以检测代码中不必要的重复，使代码更加简洁和减少冗余代码。测试代码中的单元测试的覆盖率，帮助程序员更好的修改代码。还可以帮助检测代码中潜在的 bugs，这点相信非常管用。Sonar 的安装和配置按照教程也比较简单，很快就实现了 Sonar Qube 的启动，但是对于 Sonar 所反映的各项表示代码质量的指标，还是不是很理解。不是很清楚各项指标多代表的含义。要使用 Sonar 时，先要进入 Sonar 的安装目录下，先启动 SonarQube，在进入需要测试的代码目录，写好 properties 文件，然后运行：

sonar-runner 再在 <http://localhost:9000> 下的项目管理界面查看结果就可以了，我自己测试 Calculator 时的结果如下：

```
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
Total time: 18.997s
Final Memory: 11M/233M
INFO: -----
```